Hindawi
Complexity
Volume 2018, Article ID 9826243, 15 pages
https://doi.org/10.1155/2018/9826243

Research Article

Underestimated Cost of Targeted Attacks on Complex Networks

Xiao-Long Ren,[1] **Niels Gleinig**,[2] **Dijana Tolić**,[3] **and Nino Antulov-Fantulin**[1]

[1] *Computational Social Science, ETH Zürich, Clausiusstraße 50, 8092 Zürich, Switzerland*
[2] *Department of Computer Science, ETH Zürich, Zürich, Switzerland*
[3] *Laboratory for Machine Learning and Knowledge Representations, Rudjer Bošković Institute, Zagreb, Croatia*

Correspondence should be addressed to Nino Antulov-Fantulin; nino.antulov@gess.ethz.ch

Received 25 August 2017; Revised 7 December 2017; Accepted 17 December 2017; Published 17 January 2018

Academic Editor: Ilaria Giannoccaro

The robustness of complex networks under targeted attacks is deeply connected to the resilience of complex systems, which is defined as the ability to make appropriate response to the attack. In this paper, we study robustness of complex networks under a realistic assumption that the cost of removing a node is not constant but rather proportional to the degree of a node or equivalently to the number of removed links a removal action produces. We have investigated the state-of-the-art targeted node removing algorithms and demonstrate that they become very inefficient when the cost of the attack is taken into consideration. For the case when it is possible to attack or remove links, we propose a simple and efficient edge removal strategy named Hierarchical Power Iterative Normalized cut (HPI-Ncut). The results on real and artificial networks show that the HPI-Ncut algorithm outperforms all the node removal and link removal attack algorithms when the same definition of cost is taken into consideration. In addition, we show that, on sparse networks, the complexity of this hierarchical power iteration edge removal algorithm is only $O(n\log^{2+\epsilon}n)$.

1. Introduction

The ability of complex system to dynamically adapt to internal failures or external disturbances is called a resilience. The adaptation is connected to the robustness of the network structure [1], which is defined as the ability to maintain functionality without adaptation to internal failures or external disturbances (attacks). In this paper, we will focus on the robustness of complex networks under targeted attacks with the more realistic cost function. Robustness of connected components under random failure of nodes or links is described with the classical percolation theory [2, 3]. Percolation is the simplest process showing a continuous phase transition, scale invariance, fractal structure, and universality and it is described with just a single parameter, that is, the probability of removing a node or edge. Network science studies have demonstrated that scale-free networks [4, 5] are more robust than random networks [6, 7] under random attacks or failures but less robust under targeted attacks [8–12]. Recently, studies of network resilience have moved their focus to more realistic scenarios of interdependent networks [13], competing networks [14], different failure [15], and recovery [16, 17] mechanisms.

Although the study of network robustness has received a huge amount of attention, the majority of the targeted attack strategies are still based on the heuristic identification of influential nodes [11, 18–21] with no performance guarantees for the optimality of the solution. Finding the minimal set of nodes such that their removal maximally fragments the network is called the network dismantling problem [22, 23] and it belongs to the NP-hard class. Thus no polynomial-time algorithm has been found for it and only recently different state-of-the-art methods were proposed as approximation algorithms [22–28] for this task. Although state-of-the-art methods show promising results for network dismantling, we take one step back and analyze the implicit assumption these network dismantling algorithms have. The implicit assumption that the cost of a removing action is equivalent for all nodes regardless of their importance or centrality in network is not a realistic one. Attacking a central node, for example, a high degree node in sociotechnical systems, usually comes with the higher additional cost when compared to the same action on a low degree node. Therefore, it is more realistic to explicitly assume that the cost of an attack is heterogeneous. In this paper, we define the cost of removing a node as a function of its degree.

Recently, similar definition of the cost [29] was used to analyze fragmentation and strengthening process for a class of random network models. Under the assumption of the random network models, they found that the optimal cost for fragmentation and strengthening process consists out of the list of priorities of degrees for removed nodes which is independent of the network's degree distribution.

In this work, we make the explicit assumption that the cost of an attack is proportional to the degree of a node or equivalently to the number of adjacent links a removed node has. We investigated different state-of-the-art node removal algorithms on real networks and results show that with respect to this concept of cost, most state-of-the-art algorithms are very inefficient and in most instances perform even worse than the random removal strategy for a fixed finite budget of cost.

Furthermore, when edge removal attacks are possible, we compare them to the node removal strategies with respect to the same definition of cost, that is, the number of removed links needed to fragment the network. Note that removing a node is equivalent to removing all the edges of that node, and therefore all node removal actions can be reproduced with the edge removal strategy but vice versa does not hold. Therefore, we also make highlight that the comparisons between node and edge based strategies are only interpretable in cases when edge based attacks are possible. In that case, we propose and use an edge removal strategy, named the *Hierarchical Power Iterative Normalized cut* (HPI-Ncut) as one of the possible solutions to overcome the large fragmentation cost. Although edge based strategies have higher degree of freedom as they can remove only a fraction of edges adjacent to the node, still we find cases where node-based strategies can outperform the edge based strategies. However, our proposed method (HPI-Ncut) always outperforms all the state-of-the-art targeted node-based attack algorithms and edge removal strategies [18, 27, 30].

The structure of this paper is organized as follows. First, in Section 2 ("Materials and Methods"), we introduce the empirical and artificial networks that are used in this paper (Section 2.1), present and describe current targeted attack strategies (Section 2.2), define a degree cost-fragmentation measure (Section 2.3), and describe the proposed HPI-Ncut method (Section 2.4). Then, in Section 3 ("Results and Discussions"), we quantify the cost of the state-of-the-art node removal strategies and show that in most cases the cost of such attacks is inefficient with respect to the degree-based definition of cost (Section 3.1). These results have important impact for real world scenarios of network fragmentations where cost budget is limited. Finally, when it is possible to remove single edges (e.g., shielding communication links, removing power lines, cutting off trading relationships, or others), we use the proposed HPI-Ncut method and compare its performances with other strategies (Section 3.2). The effect of edge removal HPI-Ncut method as an immunization measure for the epidemic spreading process on networks is presented (Section 3.3).

2. Materials and Methods

In this section, we describe data sets and some existing state-of-the-art targeted attack algorithms. Among them, the node removal-based attack algorithms are designed to dismantle the network into pieces with no thought for the cost of the attacking. In other words, these algorithms consider all the nodes have uniform cost. We also introduce the edge betweenness and bridgeness, which originally proposed evaluateing the importance of nodes, as two comparable link attacking methods. Then, we define the degree cost-fragmentation effectiveness (DCFE) as an index to measure the performance of different attacking methods. At last, we introduce the degree cost-fragmentation effectiveness measure and present the HPI-Ncut method.

2.1. Data Sets. To evaluate the performances of the network dismantling (fragmentation) algorithms, we used both real networks and synthetic networks in this paper: (a) *Political Blogs* [31] which is an undirected social network that was collected around the time of the US presidential election in 2004. This network is a relatively dense network whose average degree is 27.36; (b) *Petster-hamster* which is an undirected social network which contains friendships and family links between users of the website http://hamsterster.com. This network data set can be downloaded from KONECT (http://konect.uni-koblenz.de/networks/petster-hamster); (c) *Power Grid* [32] which is an undirected Power Grid network in which a node is either a generator, a transformer, or a substation, while a link represents a transmission line. This network data set can also be downloaded from KONECT (http://konect.uni-koblenz.de/networks/opsahl-powergrid); (d) *Autonomous Systems* is an undirected network from the University of Oregon Route Views Project [33]. This network data set can be downloaded from SNAP (https://snap.stanford.edu/data/as.html); (e) Erdös and Rényi (ER) network [34] is constructed with 2500 nodes. Its average degree is 20 and the connection probability is 0.01; (f) Scale-free (SF) network with size 10,000, exponent 2.5, and average degree 4.68; (g) Scale-free (SF) network with size 10,000, exponent 3.5, and average degree 2.35; (h) stochastic block model (SBM) with ten clusters is an undirected network with 4232 nodes and average degree 2.60. The basic properties of these networks are listed in Table 1.

2.2. Compared Attack Strategies. In this subsection, we will briefly introduce state-of-the-art node removal attack algorithms and some edge evaluation methods which are used in this paper. We also employ several baselines methods for edge based attacks, which are based on the random edge removal and sequential removal of edges with high betweenness and bridgeness measures.

Table 1: Basic statistical features of the GCCs of the eight real and synthetic networks.

Network	Nodes	Links	Avg. degree	Sparsity
Political Blogs (PB)	1222	16714	27.36	$2.24E-2$
Petster-hamster (PH)	2000	16098	16.10	$8.05E-3$
Power Grid (PG)	4941	6594	2.67	$5.40E-4$
Autonomous Systems (AS)	6474	12572	3.88	$6.00E-4$
ER	2500	12500	10.00	$4.00E-3$
SF ($\gamma = 2.5$)	10000	23423	4.68	$4.69E-4$
SF ($\gamma = 3.5$)	10000	11761	2.35	$2.35E-4$
SBM	4232	5503	2.60	$6.15E-4$

(i) Percolation method: in the study of the network attacks, percolation [35] is a random process of uniform removal of either nodes (site percolation) or edges (bond percolation).

(ii) High degree (HD) method [9, 36]: in HD method, all the nodes are ranked according to their degrees at the beginning. Then the highest ranked node (and its associated edges) will be removed one by one. The high degree adaptive (HDA) method is an adaptive version of the HD method. The HDA recomputes and ranks the degree of all the nodes before every removing.

(iii) Equal graph partitioning (EGP) algorithm: EGP algorithm [21], which is based on the nested dissection [37] algorithm, can partition a network into two groups with arbitrary size ratio. In every iteration, EGP algorithm divides the target nodes set into three subsets: first group, second group, and the separate group. The separate group is made up of all the nodes that connect to both the first group and the second group. Then minimize the separate group by trying to move nodes to the first group or the second group. Finally, after removing all the nodes in the separate group, the original network will be decomposed into two groups. In our implementation, we partition the network into two groups with approximate equal size.

(iv) Collective Influence (CI) algorithm: CI algorithm [24] attacks the network by mapping the integrity of a tree-like random network into optimal percolation theory [38] to identify the minimal separate set. Specifically, the collective influence of a node is computed by the degree of the neighbors belonging to the frontier of a ball with radius l. CI is an adaptive algorithm which iteratively removes the node with highest CI value after computing the CI values of all the nodes in the residual network. In our implementation, we compute the CI values with $l = 3$.

(v) Min-Sum algorithm: the three-stage Min-Sum algorithm [22] includes (1) breaking all the circles, which could be detected form the 2-core [19] of a network, by the Min-Sum message passing algorithm, (2) breaking all the trees larger than a threshold C_1, (3) greedily reinserting short cycles no greater than a threshold C_2, which ensures that the size of the GCC is not too large. In our implementation, we set C_1 and C_2 as 0.5% and 1% of the size of the networks.

(vi) CoreHD algorithm: inspired by Min-Sum algorithm, CoreHD algorithm [23] iteratively deletes the node with highest degree from the 2-core [19] of the residual network.

(vii) Belief propagation-guided decimation (BPD) [28, 39]: the BPD method is a loop-focused global algorithms which removes a set of nodes so that all the loops in the network are broken. In every iteration process, the node with the highest probability of being suitable for deletion is deleted. After the deletion of a specific fraction of nodes, the probability of all the nodes will be updated.

(viii) Edge betweenness [40]: betweenness is a widely used centrality measure which is the sum of the fraction of all-pairs shortest paths that pass a node. Edge betweenness, an extension of the betweenness, is used to evaluate the importance of a link and is defined as the sum of the fraction of all-pairs shortest paths that pass this link [36]. In this strategy, the links are removed sequentially from high to low edge betweenness value.

(ix) Bridgeness [30]: bridgeness uses local information of the network topology to evaluate the significance of edges in maintaining network connectivity. The bridgeness of a link is determined by the size of k-clique communities that the two end points of this link are connected with and the size of the k-clique communities that this link is belonging to.

2.3. Degree Cost-Fragmentation Effectiveness (DCFE). The robustness of the network structure can be measured in different means, but a common way is to characterize the function of the size of the largest connected component (GCC) with respect to the ratio of the removed nodes or edges, that is, cost. Characterization of this function was done in two distinct ways: (i) by the value of critical point when the largest component completely collapses [41] or (ii) by measuring the size of the largest component during the whole attacking process [42]. However, only recently [29] the cost functions for nodes attacks were formulated in a more general way as a function of degree.

We make explicit assumption that the cost of removing a node is proportional to the degree or to the number of the adjacent edges that have to be removed. Let us define the function $f_{\mathscr{D}}(x)$ as the size of GCC for fixed attack cost x for strategy $\mathscr{D}$. The cost $x \in [0,1]$ is measured as the ratio of the number of removed edges in the network. Now, for a fixed budget x, strategy $\mathscr{D}$ is more efficient than strategy $\mathscr{L}$ if and only if $f_{\mathscr{D}}(x) < f_{\mathscr{L}}(x)$; that is, the size of the GCC is smaller by attacking with strategy $\mathscr{D}$ than with strategy $\mathscr{L}$ with the limited budget x.

Here we define the *degree cost-fragmentation effectiveness* (DCFE) for strategy $\mathscr{D}$ as the area under the curve of the size of GCC versus the cost, which can be computed as the integral over all possible budgets: $F_{\mathscr{D}} = \int_0^1 f_{\mathscr{D}}(x)dx$. This measure is a combination of the robustness measure that takes all possible cost budgets [42] into consideration. Here, the cost is proportional to the degree [29] or to the number of the adjacent edges that have to be removed. Smaller value of DCFE implies that the attack has stronger effect over all possible budgets.

2.4. HPI-Ncut: Edge Removal Strategy. In this section, we introduce and describe the Hierarchical Power Iterative Normalized cut for edge removal strategy (HPI-Ncut). Thus, if the edge removal actions on networks are applicable, we compare them with the same definition of the cost to the node-based strategies. The link fragmentation problem can be narrated as follows: if we have a budget of x links that can be attacked or removed, which links should we pick? This is mathematically equivalent to asking how to partition a given network with a minimal separate set of edges.

We applied the spectral strategy for edge attack problem, which fall in the class of well-known spectral clustering and partitioning algorithms [43–47]. We use the hierarchical partitioning with Ncut objective function [44] combined with power iteration procedure for approximation of eigenvectors.

Now, we describe the hierarchical iterative algorithm for edge removing. This algorithm hierarchically applies the spectral bisection algorithm, which has the same objective function as the normalized cut algorithm [44]. Furthermore we have used the power iteration method to approximate spectral bisection. In order to explain our algorithm, we quickly recall the spectral bisection algorithm.

The Spectral Bisection Algorithm

Input: adjacency matrix W of a network

Output: a separated set of edges that partition the network into two disconnected clusters A, $\overline{A}$:

(1) compute the eigenvector v_2, which corresponds to the second smallest eigenvalue of the normalized Laplacian matrix $L_w = D^{-1/2}(D-W)D^{-1/2}$, or some other vector v for which $v^T L_w v / v^T v$ is close to minimal. We use the power iteration method to compute this vector, which will be explained later,

(2) put all the nodes with $v_2(i) > 0$ into the first cluster A and all the nodes with $v_2(i) \leq 0$ into the second cluster $\overline{A}$. All the edges between these two clusters form the separation set that can partition the network.

The clusters that we obtained by this method had usually very balanced sizes. If, however, it is very important to get clusters of exactly the same size, one could put those $n/2$ nodes with the largest entries in v_2 into one cluster and the remaining nodes into the other cluster.

Hierarchical Power Iterative Normalized Cut (HPI-Ncut) Algorithm

Input: adjacency matrix of a network

Output: partition of the network into small groups:

(1) partition the GCC of the network into two disconnected clusters A and $\overline{A}$ by using the spectral bisection algorithm and removing all the links in the separated set,

(2) if the budget for link removal has not been overrun and if the GCC is not yet small enough, partition A and $\overline{A}$ with Step (1), respectively.

The reason why we cluster hierarchically is because this allows us to refine the fragmentation gradually. For example, if, after partitioning the network into 2^k clusters, we decide that the clusters should be smaller, we would just have to partition each of the existing clusters into 2 new clusters, obtaining 2^{k+1} clusters. So the links that were removed already remain removed and we just need to remove some additional ones. If, however, we had used spectral clustering straightforwardly, it could happen that the set of links to be removed in order to partition the network into 2^{k+1} clusters would not contain the set of links that needed to be removed for 2^k clusters.

Power Iteration Method

Input: adjacency matrix W of a network, $\eta(n)$ number of iterations

Output: the eigenvector v_2 or some other vector v for which $v^T L_w v / v^T v$ is close to λ_2:

(1) draw v randomly with uniform distribution on the unit sphere

(2) set $v \leftarrow v - v_1^T v \cdot v_1$, where $v_1 = (1/\sqrt{n})(1, \dots, 1)$

(3) for $i = 1$ to $\eta(n)$

$v \leftarrow \widetilde{L} v / \|\widetilde{L} v\|$, where $\widetilde{L} = 2 \cdot I - L_w$ and $L_w = D^{-1/2}(D-W)D^{-1/2}$.

Objective Function of the Spectral Bisection Algorithm. In Appendix A, we show that the spectral bisection algorithm has the same objective function with the relaxed Ncut [44] algorithm:

$$\mathrm{Ncut}\left(A,\overline{A}\right) := \sum_{i\in A,j\in\overline{A}} W_{i,j}\left(\frac{1}{\sum_{i\in A} D_{ii}}+\frac{1}{\sum_{j\in\overline{A}} D_{jj}}\right), \quad (1)$$

where $A \subseteq V$ denotes set of nodes in the first partition, $\overline{A}$ denotes the set of nodes in the second partition, and D_{ii} is the degree of the node i.

The main reason we used this objective function is that it minimizes the number of links that are removed and the total sum of node degree centralities in both partitions A and $\overline{A}$ is approximately equal. In Appendix B, we show the exponential convergence of the power iteration method to the eigenvector associated with the second smallest eigenvalue of L_w.

Complexity of the HPI-Ncut Algorithm. In Appendix C, we show that the complexity of the spectral bisection algorithm is $O(\eta(n)\cdot n\cdot \overline{d})$ and the complexity of the hierarchical clustering algorithm is $O(\eta(n)\cdot n\cdot \overline{d}\cdot \log(n))$ where $\eta(n)$ is the number of iterations in the power iteration method. The power iteration method converges with exponential speed as $\eta(n) \to \infty$. The average degree $\overline{d}$ is almost constant for large sparse network. Hence we may expect asymptotically good results with $\eta(n) = \log(n)^{1+\varepsilon}$ for any $\varepsilon > 0$, giving the hierarchical spectral clustering algorithm a complexity of $O(n\cdot \log^{2+\varepsilon}(n))$. In practice, we have used $\varepsilon = 0.1$, which gives a complexity of $O(n\cdot \log^{2.1}(n))$.

3. Results and Discussions

In this section, we compare existing node targeting attack strategies with respect to the new definition of cost. We make explicit assumption that the cost of removing a node is proportional to the number of the adjacent edges that have to be removed. This suggests that the nodes with higher degree have higher associated removal cost.

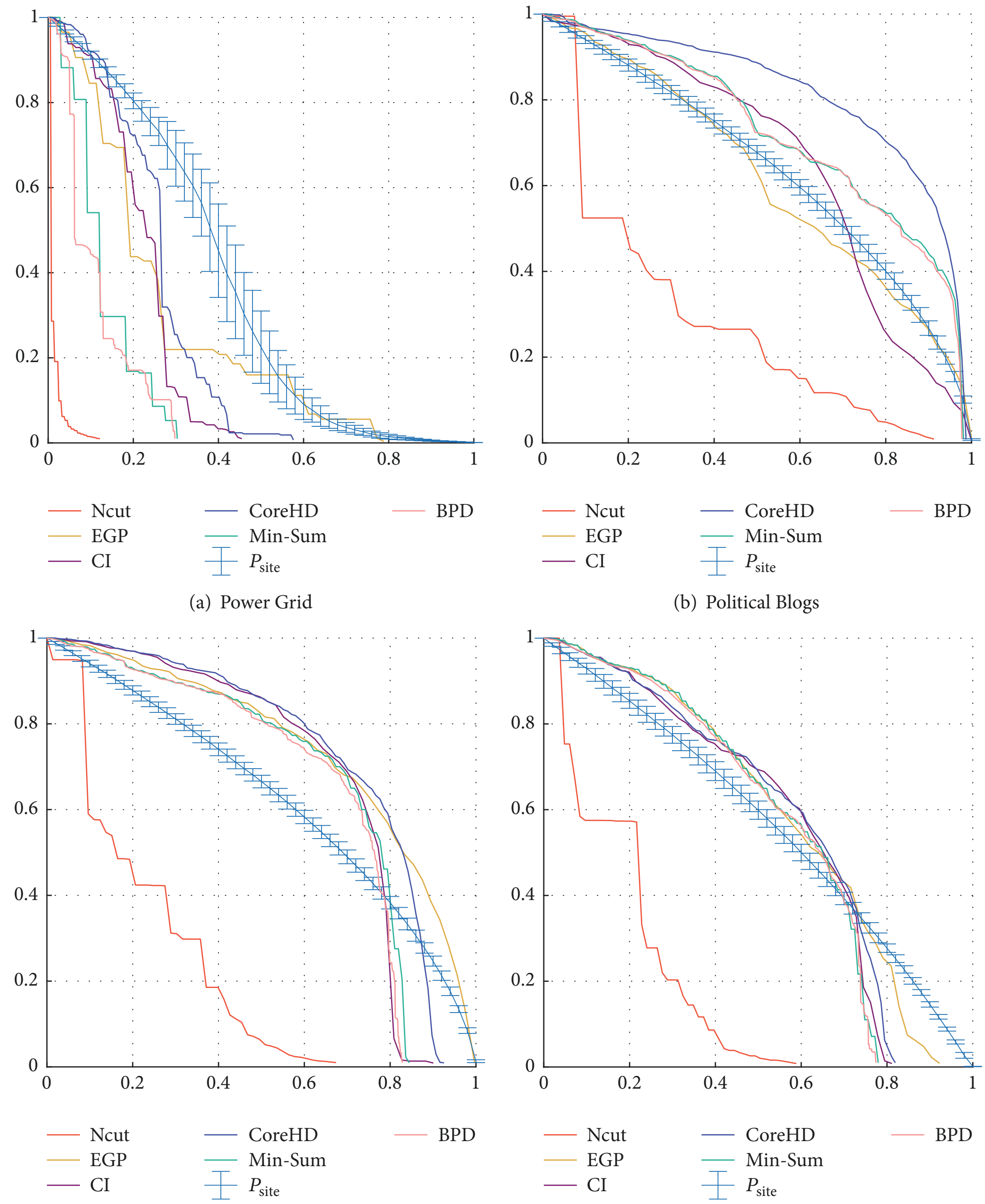
3.1. Effectiveness of the Node Targeting Attack Strategies. By taking into the account the degree-based cost in targeted attacks, the results can be highly counterintuitive. The performances of the state-of-the-art node removal-based methods are in some cases even worse than the naive process of random removal of nodes (site percolation), when we take into account the attack cost, as shown in Figures 1 and 2. In fact, networks have their intrinsic resilience under attacking for their distinct network structures. To avoid the interference of the architectural difference of networks, we use site percolation method as a baseline null model. The site percolation strategy randomly removes nodes in a network, which could be used to reflect the intrinsic resilience of the attacked network to a certain extent. The cost-fragmentation effectiveness of the site percolation is denoted with $F_* = \int_0^1 f_*(x)dx$; see details in Section 2.3.

Table 2 summarizes the DCFE of different attack strategies on eight networks. Table 3 summarizes the improvement of DCFE of different attack strategies $\mathscr{D}$ comparing with the null model (site percolation), which is calculated as $\int_0^1 (f_*(x) - f_{\mathscr{D}}(x))dx$. On the whole, all node-centric strategies (HD, HDA, EGP, CI, CoreHD, Min-Sum, and BPD) distinctly work better than baseline on the three networks with lower average degree, that is, Power Grid, SF ($\gamma = 3.5$), and SBM network. However, on empirical social Petster-hamster network, Political Blogs network, Autonomous Systems network, and SF ($\gamma = 2.5$) network, all these node-centric strategies are comparably equal or even worse than the baseline method, according to the DCFE score. More interestingly, for a fixed budget, many networks are more fragile with the HD attack strategy than by HDA, as the results shown in Tables 2 and 3. In the last line of Table 3, we compute the average value of the improvement over different networks, which can reflect the overall performance of the algorithms. These results suggest that state-of-the-art node removal-based algorithms in realistic settings are rather inefficient if the cost of fragmentation is taken into account.

3.2. Effectiveness of the HPI-Ncut Attacks. In this section, we will compare the proposed edge removal-based attack strategy, HPI-Ncut algorithm, with random uniform attack, edge betweenness, bridgeness, and some classical node removing strategies (see the details in Section 2). The results show that the HPI-Ncut strategy greatly decreases the cost of the attack, comparing with the state-of-the-art removing strategies.

In general case, each attack strategy algorithm could generate a ranking list of all (or partial) nodes or links of the network. After removing the nodes or links one after another, the size of the GCC of the residual network characterizes the effectiveness of each algorithm. The removal process will cease when the size of the GCC is smaller than a given threshold (here we use 0.01). In this paper, to test the effectiveness of this spectral edge removal algorithm, HPI-Ncut, we plot the size of the GCC versus the removal fraction of links, for both real networks (Figures 1 and 3) and synthetic networks (Figures 2 and 4), comparing with classical node removing algorithms (Figures 1 and 2) and existing link evaluation methods (Figures 3 and 4). The results show that the HPI-Ncut algorithm outperforms all the other attack algorithms.

In Figures 1 and 2, we compared the HPI-Ncut algorithm with some state-of-the-art node removal-based target attack algorithms. Figure 1(a) shows that all the node removal-based algorithms are better than the site percolation method on Power Grid network, which is because the average degree of the Power Grid network is very low, only 2.67. This could also be verified by the results in Figures 2(c) and 2(d), in which the average degree of the SF ($\gamma = 3.5$) and the SBM network are 2.35 and 2.60, respectively. The trends of the curves in Figures 1 and 2 also show that the target attack algorithms work better on networks with lower average degree. Furthermore, regardless of the HPI-Ncut algorithm, other algorithms have poorer performance than baseline method (site percolation). The performances of site percolation are better until the proportion of the removed links is greater than 0.7 on SF ($\gamma = 3.5$) network and until the proportion is greater than 0.2 on SF ($\gamma = 2.5$) network. The site percolation on the SF



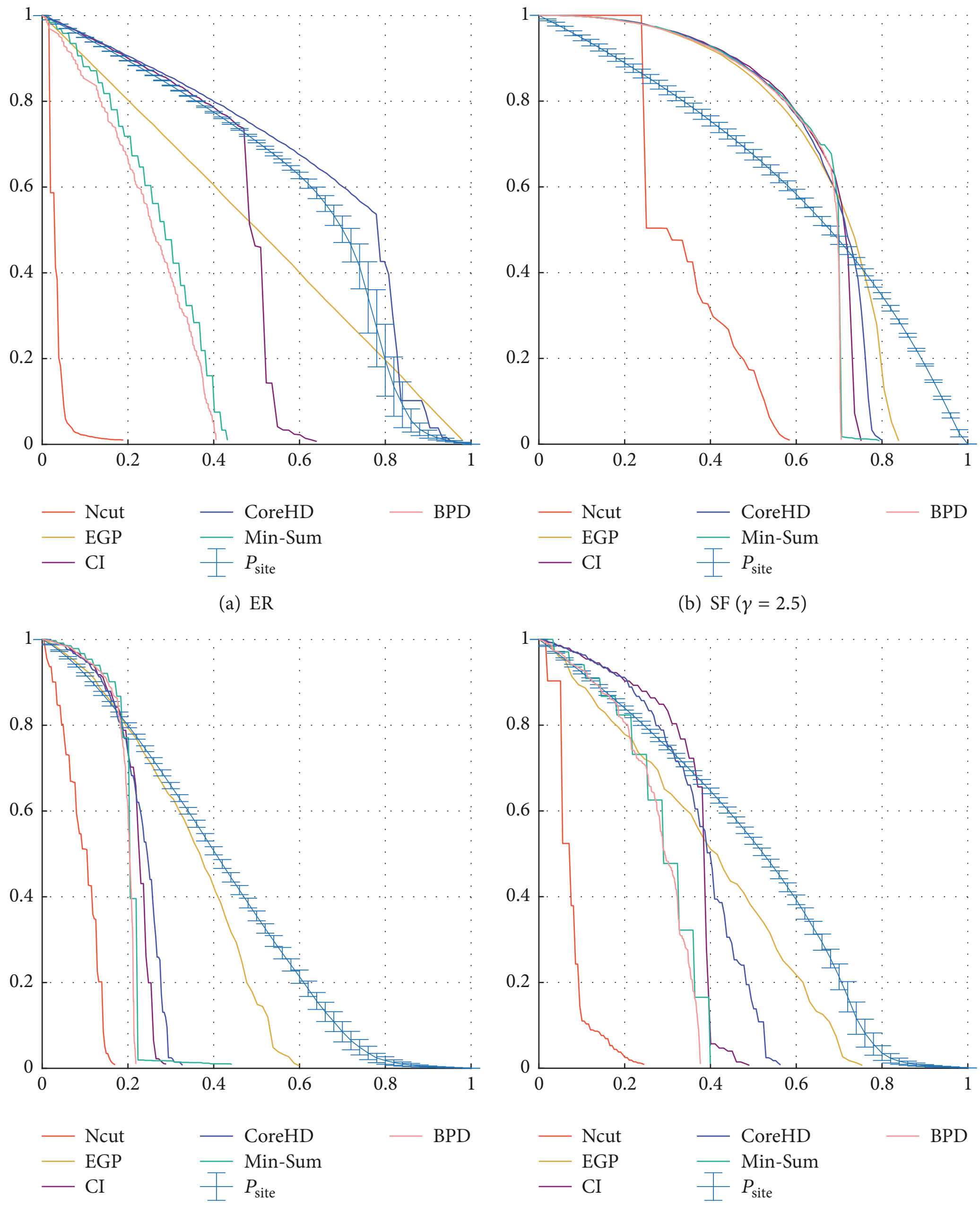
(a) Power Grid (b) Political Blogs

(c) Petster-hamster (d) Autonomous Systems

Figure 1: The size of the GCC of the networks versus the link removing proportion, comparing with classical node removal-based methods on real networks. The results of the site percolation are obtained after 100 independent runs.

Table 2: DCFE, that is, the area under the curve of the size of the GCC after attacking by different algorithms. P_{site} is short for site percolation, P_{bond} for bond percolation, Betw for betweenness, and Bridg for bridgeness. The best performing algorithm in each row is emphasized in bold.

DCFE	P_{site}	HD	HDA	EGP	CI	Min-Sum	CoreHD	BPD	P_{bond}	Betw	Bridg	HPI-Ncut
PB	0.638	0.920	0.861	0.619	0.657	0.726	0.815	0.722	0.843	0.597	0.910	**0.278**
PH	0.627	0.677	0.696	0.747	0.687	0.675	0.736	0.661	0.817	0.536	0.689	**0.224**
PG	0.371	0.260	0.293	0.263	0.219	0.130	0.256	0.113	0.305	0.145	0.420	**0.014**
AS	0.567	0.576	0.604	0.592	0.567	0.567	0.576	0.561	0.605	0.527	0.618	**0.192**
ER	0.601	0.547	0.647	0.502	0.441	0.268	0.647	0.247	0.753	0.387	0.542	**0.032**
SF ($\gamma = 2.5$)	0.619	0.700	0.706	0.671	0.650	0.636	0.660	0.632	0.683	0.672	0.694	**0.342**
SF ($\gamma = 3.5$)	0.406	0.231	0.228	0.343	0.214	0.202	0.227	0.192	0.298	0.312	0.352	**0.092**
SBM	0.487	0.419	0.378	0.397	0.348	0.284	0.374	0.274	0.384	0.348	0.512	**0.075**



(a) ER (b) SF ($\gamma = 2.5$) (c) SF ($\gamma = 3.5$) (d) SBM with ten clusters

Figure 2: The size of the GCC of the networks versus link removing proportion, comparing with classical node removal-based methods on artificial networks. The results of the site percolation are obtained after 100 independent runs.

Table 3: The improvement of the DCFE of each algorithm, comparing with the baseline, that is, site percolation method. The best performing algorithm in each column is emphasized in bold.

Improvement	P_{bond}	HD	HDA	EGP	CI	Min-Sum	CoreHD	BPD	Betw	Bridg	HPI-Ncut
PB	−32%	−44%	−35%	3%	−3%	−14%	−28%	−13%	6%	−43%	**56%**
PH	−30%	−8%	−11%	−19%	−10%	−8%	−17%	−5%	15%	−10%	**64%**
PG	18%	30%	21%	29%	41%	65%	31%	70%	61%	−13%	**96%**
AS	−7%	−2%	−7%	−4%	0	0	−2%	1%	7%	−9%	**66%**
ER	−25%	9%	−8%	17%	27%	55%	−8%	59%	36%	10%	**95%**
$SF_{\gamma=2.5}$	−10%	−13%	−14%	−8%	−5%	−3%	−7%	−2%	−9%	−12%	**45%**
$SF_{\gamma=3.5}$	27%	43%	44%	15%	47%	50%	44%	53%	23%	13%	**77%**
SBM	20%	13%	22%	18%	28%	41%	23%	44%	28%	−6%	**84%**
Average	−5%	3%	2%	6%	16%	23%	5%	26%	21%	−9%	**73%**

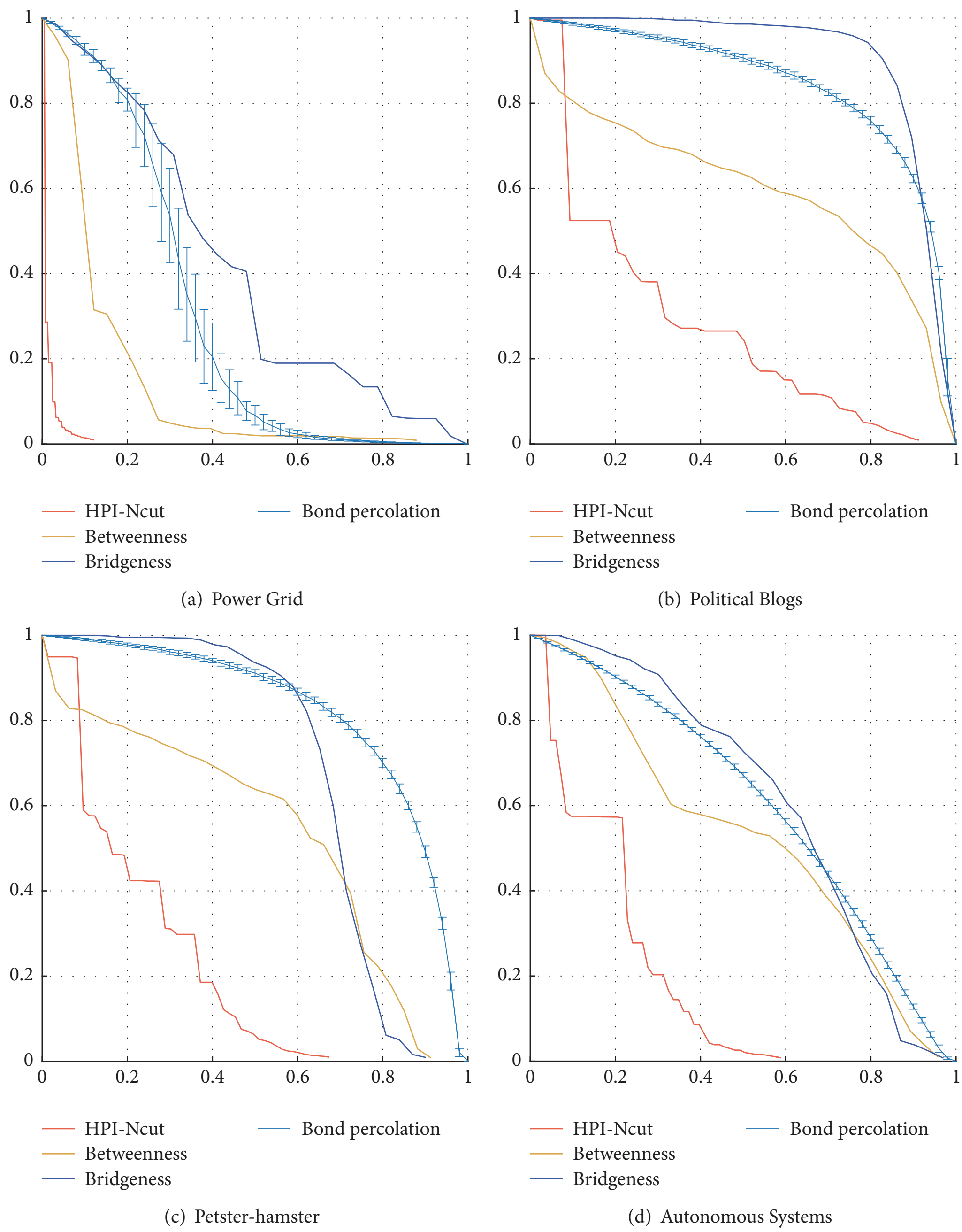
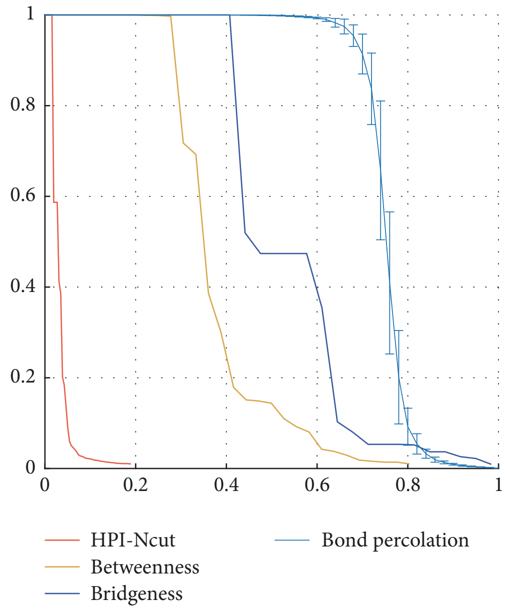


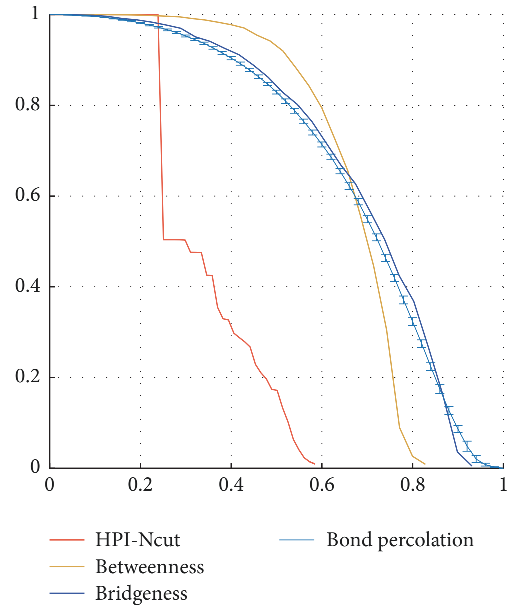
FIGURE 3: The size of the GCC of the networks versus link removing proportion, comparing with existing link removal-based methods on real networks. The results of the bond percolation are obtained after 100 independent runs.

($\gamma = 3.5$) presents an obvious phase transition phenomenon [48] comparing with the result on the SF ($\gamma = 2.5$). In addition, in Figures 2(a) and 2(d), the SBM network has obvious clusters structure comparing with the ER network. The BPD, Min-Sum, CI, CoreHD, EGP, and site percolation algorithms have a better performance on the SBM network. Moreover, the error of the site percolation method on the ER network is larger than the error on SBM network. That implies that the cluster structure of a network has a big influence on the performance of the attack strategies.

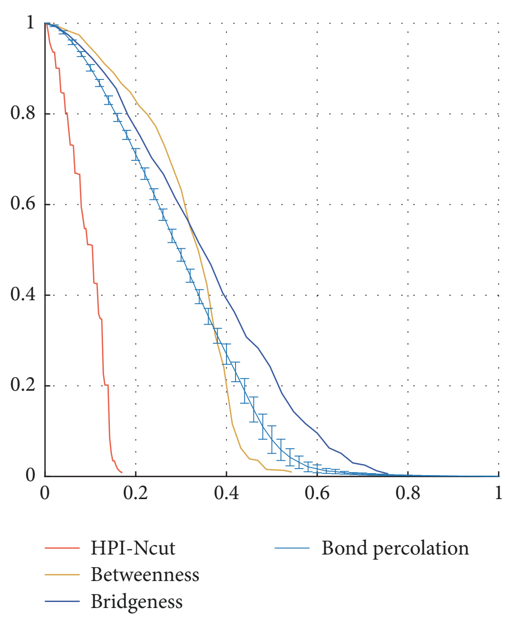
To conclude the results of Figures 1 and 2, the state-of-the-art targeted node removal strategies make large cost for optimized targeted attacks. When it is possible to apply edge-bases strategies, the HPI-Ncut algorithm overwhelmingly outperforms all the node removal-based attack algorithms, no matter on sparse or dense networks or on the networks with or without clusters structure. It is also interesting to show that some of the node targeted attack strategies (BPD, Min-Sum) can also outperform edge based strategies on several networks (PG, ER, SF, and SBM), but not the HPI-Ncut.



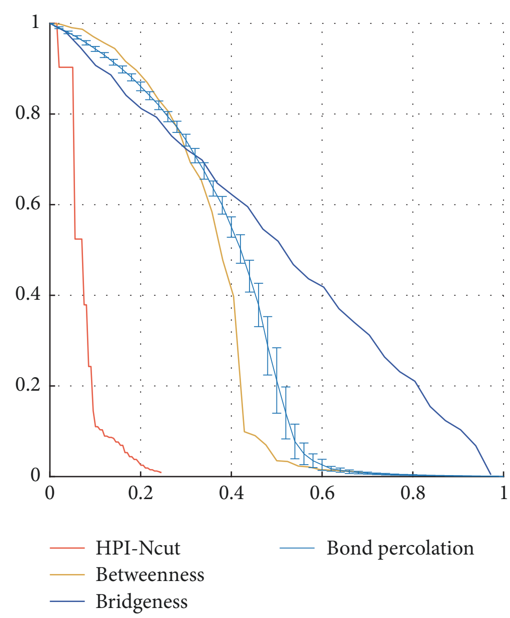
(a) ER



(b) SF ($\gamma = 2.5$)



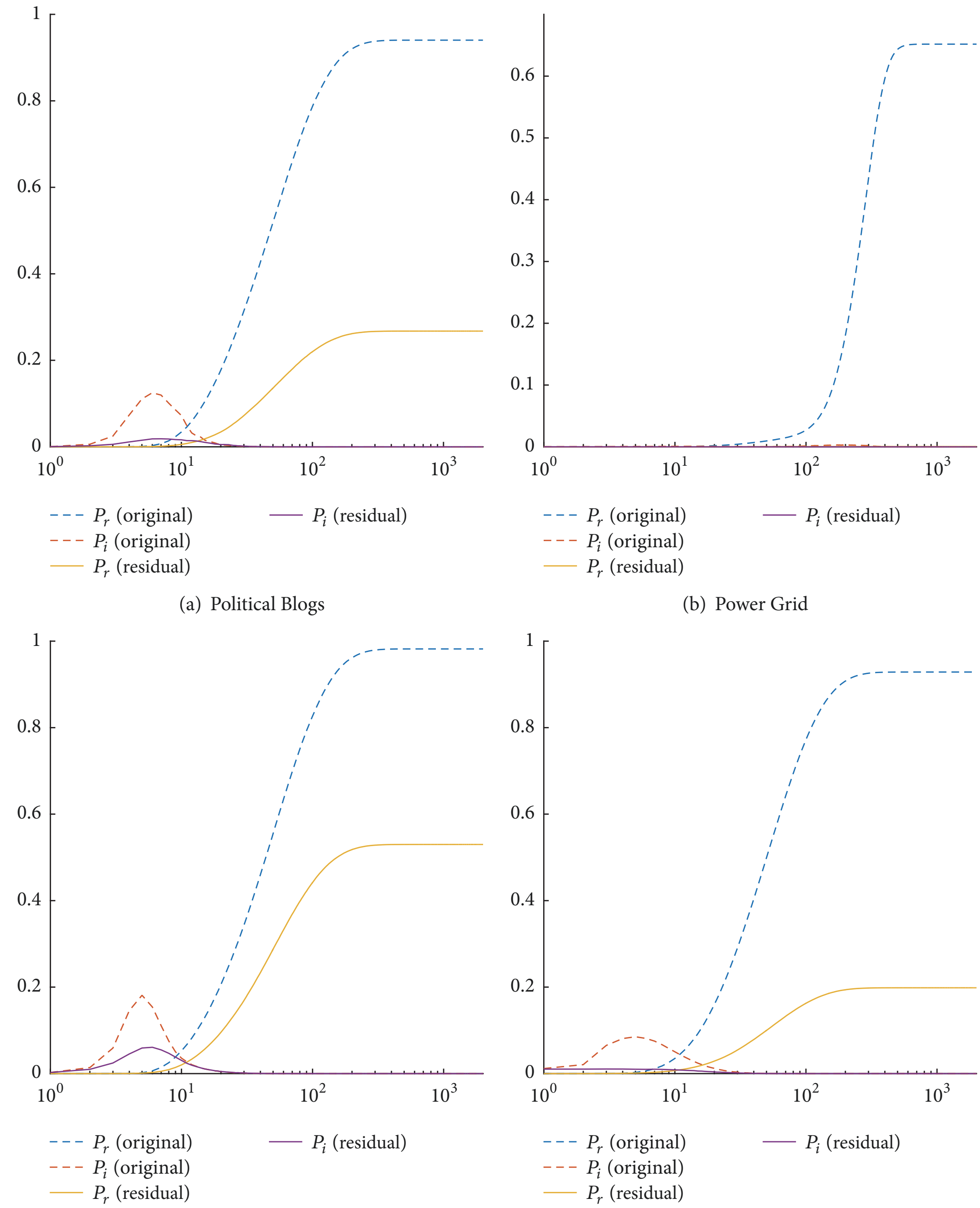
(c) SF ($\gamma = 3.5$)



(d) SBM with ten clusters

FIGURE 4: The size of the GCC of the networks versus link removing proportion, comparing with existed link removal-based methods on artificial networks. The results of the bond percolation are obtained after 100 independent runs.

In Figures 3 and 4, we compared the HPI-Ncut algorithm with some exited link evaluation algorithms. First of all, we can find that the HPI-Ncut algorithm works better and is more stable than all the other algorithms. Secondly, comparing with the results of site and bond percolation in Figures 1 and 2, we can see that the bond percolation method outperforms the site percolation method only when the average degree of the network is lower (see the results of the Power Grid, SF ($\gamma = 3.5$), and SBM network); otherwise, the site percolation is a better choice. Thirdly, in Figures 4(b) and 4(c), we can see that the bond percolation method has a better performance comparing with the edge betweenness and bridgeness algorithm when the cost is limited on scale-free networks; that is, the proportion of the removed links is smaller than 0.63 in Figure 4(b) and is smaller than 0.4 in Figure 4(c). To conclude, the HPI-Ncut algorithm overwhelmingly outperforms all the node removal-based attack algorithms and link evaluation algorithms, no matter on sparse or dense networks or on networks with or without clusters structure.



(a) Political Blogs (b) Power Grid (c) Petster-hamster (d) Autonomous Systems

Figure 5: The spreadability of the networks before and after the removing of 10% edges by HPI-Ncut algorithm. The x-axis is the time units. P_i is the number of infected entities and P_r is the number of recovered entities in the network. In the SIR model, the infection rate β is 0.10, the recovery rate is 0.02, and the basic reproduction number is 5. All the results are the average of 100 times independent runs. It is worth noting that the size of GCC of the Power Grid network is only 54 after removing 10% of links by HPI-Ncut algorithm.

3.3. Spreading Dynamics after HPI-Ncut Immunization. To more intuitively display the ability of the HPI-Ncut to make immunization of links, we studied the susceptible-infectious-recovery (SIR) [49] epidemic spreading process on four real networks. We compared both the spreading speed and spreading scope on these networks before and after targeted immunization by HPI-Ncut. The simulation results in Figure 5 show that, by simply removing 10% of links, the function of the networks had been profoundly affected by the HPI-Ncut immunization. The proportion of the GCC of the Political Blogs, Power Grid, Petster-hamster, and Autonomous Systems network after attack are 37% (449/1222), 1% (54/4941), 57% (1146/2000), and 37% (2387/6474), respectively. Thus, the spreading speeds are greatly delayed and the spreading scoops are tremendously shrunken on these networks.

4. Conclusion

To summarize, we investigated some state-of-the-art node target attack algorithms and found that they are very inefficient when the degree-based cost of the attack is taken into consideration. The cost of removing a node is defined as the number of links that have to be removed in the attack process.

We found some highly counterintuitive results; that is, the performances of the state-of-the-art node removal-based methods are even worse than the naive site percolation method with respect to the limited cost. This demonstrates that the current state-of-the-art node targeted attack strategies underestimate the heterogeneity of the cost associated with the nodes in complex networks.

Furthermore, in cases when the link removal strategies are possible, we compared the performances of the node-centric (HD, HDA, EGP, CI, CoreHD, BPD, and Min-Sum) and edge removal strategies (edge betweenness and bridgeness strategy) based on the cost of their attacks, which are measured in the same units, that is, the ratio of the removed links. We propose a hierarchical power iterative algorithm (HPI-Ncut) to fragment a network, which has the same objective function with the Ncut [44] spectral clustering algorithm. The results show that HPI-Ncut algorithm outperforms all the node removal-based attack algorithms and link evaluation algorithms on all the networks. In addition, the total complexity of the HPI-Ncut algorithm is only $O(n \cdot \log^{2+\varepsilon}(n))$, which makes it very practical to be applied on large scale networks over a million of nodes.

The underestimated cost of current state-of-the-art algorithms with respect to the degree-based cost has high influence on the development and design of better robustness and resilience mechanisms in complex systems. Furthermore, more accurate estimation of robustness under realistic conditions will allow better allocation of response resources.

Appendix

A. Objective Function

Let $G = (V, E)$ be an undirected graph with adjacency matrix W and diagonal degree matrix D, whose ith entry $D_{ii} = \sum_{j=1}^{n} W_{ij}$ is the degree of the node i. For $A \subseteq V$, let $\operatorname{cut}(A)$ denote the number of links between A and its complement $\overline{A}$. We define

$$\operatorname{Ncut}\left(A, \overline{A}\right) := \operatorname{cut}\left(A, \overline{A}\right)\left(\frac{1}{\operatorname{assoc}(A)} + \frac{1}{\operatorname{assoc}\left(\overline{A}\right)}\right), \tag{A.1}$$

where $\operatorname{assoc}(A) = \sum_{i \in A} D_{ii}$. If we describe the set A by the normalized indicator vector

$$x_A(i) := \begin{cases} 1 & \text{if } i \in A, \\ -\dfrac{\sum_{j \in A} D_{jj}}{\sum_{j \in B} D_{jj}} & \text{otherwise,} \end{cases} \tag{A.2}$$

one can show [44] that

$$\min_{A \subseteq V} \operatorname{Ncut}\left(A, \overline{A}\right) = \min_{A \subseteq V} \frac{x_A^T (D - W) x_A}{x_A^T D x_A}. \tag{A.3}$$

From the definition of Ncut one can see that finding a set A which minimizes $\operatorname{Ncut}(A, \overline{A})$ corresponds to partitioning the network into two sets A and $\overline{A}$ such that

(1) $\operatorname{cut}(A, \overline{A})$ is small and hence there are only few links between A and $\overline{A}$,

(2) $(1/\operatorname{assoc}(A) + 1/\operatorname{assoc}(\overline{A}))$ is small and so sets A and $\overline{A}$ contain more or less equally many links.

Finding such a set A is NP-hard [44], but by relaxing the constraints in the RHS of the identity (A.3) one can find good approximate solutions $\widetilde{A}$:

(1) Find

$$x_{\text{relaxed}} := \arg \min_{\substack{x \in \mathbb{R}^n, x \neq 0 \\ x^T D \vec{1} = 0}} \frac{x^T (D - W) x}{x^T D x}, \tag{A.4}$$

where we have imposed the condition $x^T D \vec{1} = 0$, because every set A for which x_A is nontrivial satisfies $x_A^T D \vec{1} = 0$.

(2) Set

$$\chi_{\widetilde{A}}(i) = \operatorname{round}\left(x_{\text{relaxed}}(i)\right) := \begin{cases} 1 & \text{if } x_{\text{relaxed}}(i) \geq 0, \\ -1 & \text{otherwise} \end{cases} \tag{A.5}$$

and define $\widetilde{A} = \{i \in \text{Nodes} \mid \chi_{\widetilde{A}}(i) = 1\}$.

The idea behind this method is that $\chi_{\widetilde{A}}$ will be the best approximation of x_{relaxed}, out of the set of all vectors with entries in -1 and 1, and since x_{relaxed} minimizes $x^T(D - W)x / x^T D x$,

$$\operatorname{Ncut}\left(\widetilde{A}\right) = \frac{x_{\widetilde{A}}^T (D - W) x_{\widetilde{A}}}{x_{\widetilde{A}}^T D x_{\widetilde{A}}} \approx \frac{\chi_{\widetilde{A}}^T (D - W) \chi_{\widetilde{A}}}{\chi_{\widetilde{A}}^T D \chi_{\widetilde{A}}} \tag{A.6}$$

will be also close to

$$\min_{A \subseteq \text{Nodes}} \operatorname{Ncut}(A) = \min_{A \subseteq \text{Nodes}} \frac{x_A^T (D - W) x_A}{x_A^T D x_A}. \tag{A.7}$$

One can show that a solution to (A.4) is given by $x_{\text{relaxed}} = D^{1/2} v_2$, where v_2 is the eigenvector of the second smallest eigenvalue λ_2 of the *normalized Laplacian matrix*

$$L_w = D^{-1/2} (D - W) D^{-1/2}. \tag{A.8}$$

D is a diagonal matrix and if the network is connected we have $D_{ii} > 0$. So the entries of the vectors x_{relaxed} and v_2 have the same sign and therefore we have $\operatorname{round}(x_{\text{relaxed}}) = \operatorname{round}(v_2)$.

B. Exponential Convergence of the Power Iteration Method

L_w is real and symmetric. Therefore it has real eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ corresponding to eigenvectors $v_1, \ldots, v_n$ which form an orthonormal basis of $\mathbb{R}^n$. One can easily show that $\lambda_1 = 0$ and $\lambda_n \leq 2$. So in order to compute v_2 we consider the matrix $\widetilde{L} = 2 \cdot I - L_w$, which has the same eigenvectors $v_1, \ldots, v_n$ as L_w. Now the corresponding eigenvalues are $\widetilde{\lambda_1} = 2 \geq \cdots \geq \widetilde{\lambda_n} = d_{\max} - \lambda_n \geq 0$ and in particular v_1 corresponds to the largest eigenvalue and v_2 to the second largest eigenvalue.

If v is a random vector uniformly drawn from the unit sphere and we force it to be perpendicular to v_1 by setting $v \leftarrow v - v_1^T v \cdot v_1$; then $v = \psi_2 v_2 + \cdots + \psi_n v_n$ and $\psi_2 \neq 0$ almost surely. Furthermore $\widetilde{L} v = \widetilde{\lambda_2} \psi_2 v_2 + \cdots + \widetilde{\lambda_n} \psi_n v_n$ and if we set $v^{(k)} := \widetilde{L}^k v$, then

$$\begin{aligned} \frac{v^{(k)}}{\left\| v^{(k)} \right\|} &= \frac{\widetilde{\lambda_2}^k \psi_2 v_2 + \cdots + \widetilde{\lambda_n}^k \psi_n v_n}{\left\| \widetilde{\lambda_2}^k \psi_2 v_2 + \cdots + \widetilde{\lambda_n}^k \psi_n v_n \right\|} \\ &= \frac{\psi_2 v_2 + \left(\widetilde{\lambda_3}/\widetilde{\lambda_2}\right)^k \psi_3 v_3 + \cdots + \left(\widetilde{\lambda_n}/\widetilde{\lambda_2}\right)^k \psi_n v_n}{\left\| \psi_2 v_2 + \left(\widetilde{\lambda_3}/\widetilde{\lambda_2}\right)^k \psi_3 v_3 + \cdots + \left(\widetilde{\lambda_n}/\widetilde{\lambda_2}\right)^k \psi_n v_n \right\|} \end{aligned} \tag{B.1}$$

converges with exponential speed to some eigenvector of L with eigenvalue λ_2, because for every i with $\lambda_i > \lambda_2$ we have $\widetilde{\lambda_i}/\widetilde{\lambda_2} < 1$ and therefore $(\widetilde{\lambda_i}/\widetilde{\lambda_2})^k \psi_i v_i \to 0$. Generally one can deduce from (B.1) that

$$\begin{aligned} &\frac{v^{(k)T} L_w v^{(k)}}{v^{(k)T} v^{(k)}} \\ &= \frac{\lambda_2 \left|\psi_2\right|^2 + \lambda_3 \left| \left(\widetilde{\lambda_3}/\widetilde{\lambda_2}\right)^k \psi_3 \right|^2 + \cdots + \lambda_n \left| \left(\widetilde{\lambda_n}/\widetilde{\lambda_2}\right)^k \psi_n \right|^2}{\left|\psi_2\right|^2 + \left| \left(\widetilde{\lambda_3}/\widetilde{\lambda_2}\right)^k \psi_3 \right|^2 + \cdots + \left| \left(\widetilde{\lambda_n}/\widetilde{\lambda_2}\right)^k \psi_n \right|^2}, \end{aligned} \tag{B.2}$$

and therefore this quantity converges to λ_2 with exponential speed.

C. Complexity

The complexity of the spectral bisection algorithm is the same as the complexity of the power iteration method. The complexity of the power iteration method equals the number of iterations $\eta(n)$ times the complexity of multiplying $\widetilde{L}$ and v, that is, $O(\eta(n) \cdot n \cdot \overline{d})$ where $\overline{d}$ is the average degree of the network, or equivalently $O(|E| \cdot \eta(n))$ where $|E|$ is the number of edges.

Assuming that the spectral bisection algorithm always produces clusters of equal size, the complexity of the hierarchical spectral clustering algorithm is then given by the sum of

(i) the complexity of applying spectral bisection once on the whole network $\rightarrow O(\eta(n) \cdot n \cdot \overline{d})$

(ii) The complexity of applying it on each of the two clusters that we obtained from the first application of spectral bisection and which will have size $n/2$

(iii) The complexity of applying it on each of the 4 clusters that we obtained from the previous step and which will have size $n/4$

(iv) The complexity of applying it on each of the $n/2 = 2^{\log_2(n)-1}$ clusters that we obtained from the previous step and which will have size $n/2^{\log_2(n)-1} = 2$.

That is, in total at most

$$\begin{aligned} &O\left(\eta(n) \cdot n \cdot \overline{d}\right) + 2 \cdot O\left(\eta(n) \cdot \frac{n}{2} \cdot \overline{d}\right) + \cdots \\ &\quad + 2^{\log_2(n)-1} \cdot O\left(\eta(n) \cdot \frac{n}{2^{\log_2(n)-1}} \cdot \overline{d}\right) \\ &\quad = \sum_{i=0}^{\log_2(n)-1} 2^i \cdot O\left(\eta(n) \cdot \frac{n}{2^i} \cdot \overline{d}\right) \\ &\quad = O\left(\eta(n) \cdot n \cdot \overline{d}\right) \sum_{i=0}^{\log_2(n)-1} 1 \\ &\quad = O\left(\eta(n) \cdot n \cdot \overline{d} \cdot \log_2(n)\right), \end{aligned} \tag{C.1}$$

where we have made the pessimistic assumption that the number of iterations and the average degrees are in each step as large as they were in the beginning.

The choice of the function $\eta(n)$ is a little bit involved. If the initial random choice of the vector v is very unfortunate, there may be many iterations needed in order to have a good approximation of the eigenvector v_2. In fact, if $\psi_2 = 0$, then this algorithm would not converge to v_2 at all; however this event has probability 0.

Another condition that might slow down the computation of v_2 is if some of the other eigenvalues λ_i, $i \geq 3$, are close to λ_2. In that case $\widetilde{\lambda_i}/\widetilde{\lambda_2}$ would be close to 1 and therefore one can see from (B.1) that the corresponding v_i might have a large contribution in $v^{(k)}$ for a long time. However when λ_i is close to λ_2, this also implies that

$$\frac{v_i^T L_w v_i}{v_i^T v_i} = \lambda_i \tag{C.2}$$

is close to

$$\min_{\|v\| \neq 0} \frac{v^T L_w \mathrm{v}}{v^T v} = \lambda_2 \tag{C.3}$$

and therefore also provides a good partition of the network, since these are the quantities that are related to the cut-size.

Due to this fast convergence, one can expect asymptotically good partitions when $\eta(n) = \log(n)^{1+\varepsilon}$ and $\varepsilon > 0$, giving the hierarchical spectral clustering algorithm a complexity of $O(n \cdot \overline{d} \cdot \log^{2+\varepsilon}(n))$ in general and $O(n \cdot \log^{2+\varepsilon}(n))$ for sparse networks.

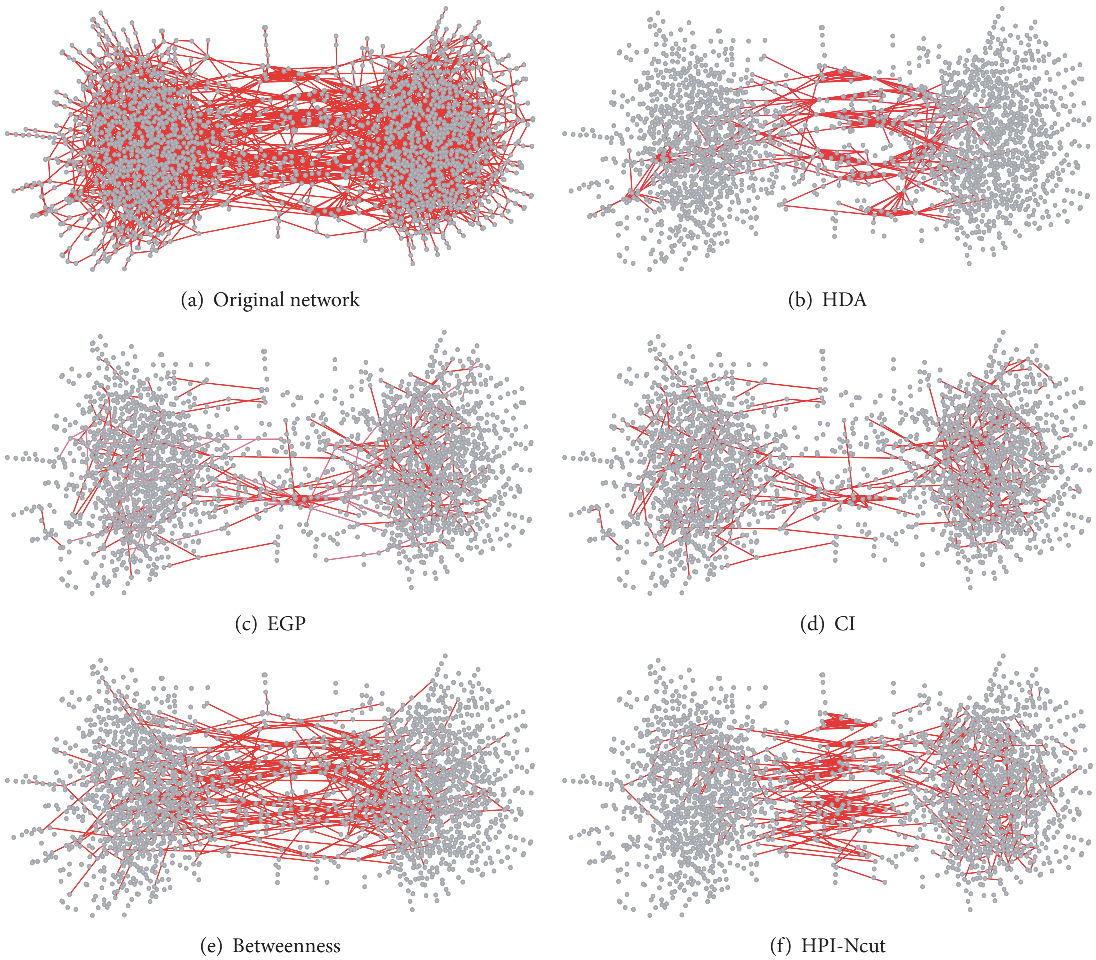


Figure 6: The schematic diagram of the removed links in a SBM network with two clusters. (a) is the original network with all the links. (b)–(f) are the top 10% links (i.e., 373 links) removed by different algorithms.

D. HPI-Ncut Algorithm with Different Number of Partitions

Previous sections give us a clear picture about the performances of different attack algorithms. Some algorithms work quite well, such as HPI-Ncut algorithm, Min-Sum algorithm, and edge betweenness algorithm, while others are not. What causes such a difference? Figure 6 may give us a clue. In this toy example, the original network is a two clusters' SBM model with totally 2078 nodes and 3729 links. Figure 6 shows the visualization of the top 10% removed links of different algorithms. Please note that the number of the red links in Figures 6(b)–6(f) is the same, namely, 373. However, comparing with edge betweenness and HPI-Ncut algorithm, much less of links between the two clusters are removed by EGP and CI algorithm, and more links are distributed among the left or the right cluster. Furthermore, comparing with edge betweenness algorithm, the links removed by HPI-Ncut algorithm mainly are distributed in the bridge part of the two clusters. This helps to partition the network into two disconnected clusters.

In the previous sections, the default target number of the disconnected clusters in HPI-Ncut algorithm is set to 2. Figure 7 shows the size of the GCC after targeted attack by HPI-Ncut with different target number of disconnected clusters, on the SBM network with two clusters and with ten clusters, respectively. Figure 7 indicates that when the original networks contains less clusters, the target number of clusters in HPI-Ncut will greatly affect the size of GCC in the initial stage of the target attack, while this influence will decline sharply in the later part of the attack process. However, the target number has a smaller impact on the attack performances of the HPI-Ncut when the original network contains much more clusters. Furthermore, when the target number of the disconnected clusters is set to 2, we can always obtain the optimal outcome on both networks. To conclude, we recommend setting the default target number of the disconnected clusters to 2 in HPI-Ncut algorithm.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

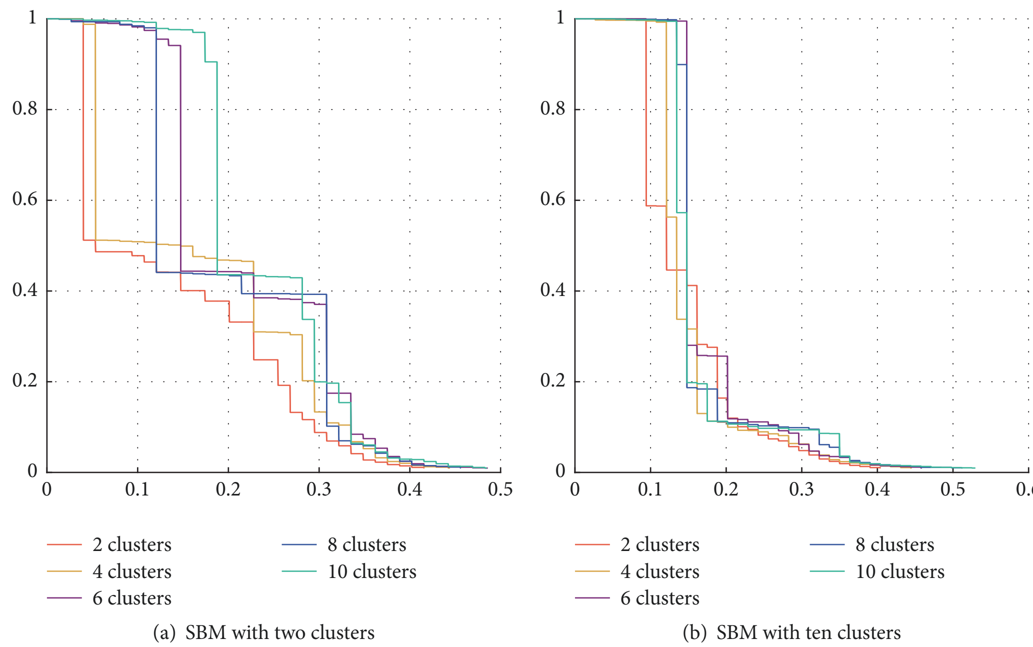


FIGURE 7: The size of the GCC of the networks versus link removing proportion, comparing of different quantities of target disconnected clusters in HPI-Ncut algorithm.

Acknowledgments

The work of Nino Antulov-Fantulin has been funded by the EU Horizon 2020 SoBigData project under Grant Agreement no. 654024. The work of Dijana Tolić is funded by the Croatian Science Foundation IP-2013-11-9623 "Machine Learning Algorithms for Insightful Analysis of Complex Data Structures." Xiao-Long Ren acknowledges the support from China Scholarship Council (CSC).

References

[1] M. E. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[2] M. Basta, V. Picciarelli, and R. Stella, "An introduction to percolation," *European Journal of Physics*, vol. 15, no. 3, article no. 001, pp. 97–101, 1994.

[3] J. P. Sethna, *Statistical Mechanics: Entropy, Order Parameters, and Complexity*, Oxford Master Series in Physics, Oxford University Press, Oxford, UK, 2006.

[4] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[5] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, "Structure of growing networks with preferential linking," *Physical Review Letters*, vol. 85, no. 21, pp. 4633–4636, 2000.

[6] P. Erdos and A. Rényi, "On the evolution of random graphs," in *Publication of the Mathematical Institute of the Hungarian Academy of Science*, pp. 17–61, 1960.

[7] E. N. Gilbert, "Random graphs," *Annals of Mathematical Statistics*, vol. 30, no. 4, pp. 1141–1144, 1959.

[8] M. Molloy and B. Reed, "A critical point for random graphs with a given degree sequence," *Random Structures & Algorithms*, vol. 6, no. 2-3, pp. 161–180, 1995.

[9] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[10] R. Cohen, K. Erez, D. Ben-Avraham, and S. Havlin, "Resilience of the Internet to random breakdowns," *Physical Review Letters*, vol. 85, no. 21, pp. 4626–4628, 2000.

[11] R. Cohen, K. Erez, D. Ben-Avraham, and S. Havlin, "Breakdown of the internet under intentional attack," *Physical Review Letters*, vol. 86, no. 16, pp. 3682–3685, 2001.

[12] T. Tanizawa, G. Paul, R. Cohen, S. Havlin, and H. E. Stanley, "Optimization of network robustness to waves of targeted and random attacks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 71, no. 4, Article ID 047101, 2005.

[13] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, pp. 1025–1028, 2010.

[14] B. Podobnik, D. Horvatic, T. Lipic, M. Perc, J. M. Buldu, and H. E. Stanley, "The cost of attack in competing networks," *Journal of the Royal Society Interface*, vol. 12, Article ID 20150770, 2015.

[15] J. Gao, X. Liu, D. Li, and S. Havlin, "Recent progress on the resilience of complex networks," *Energies*, vol. 8, no. 10, pp. 12187–12210, 2015.

[16] L. M. Shekhtman, M. M. Danziger, and S. Havlin, "Recent advances on failure and recovery in networks of networks," *Chaos, Solitons & Fractals*, vol. 90, pp. 28–36, 2015.

[17] L. Böttcher, M. Luković, J. Nagler, S. Havlin, and H. J. Herrmann, "Failure and recovery in dynamical networks," *Scientific Reports*, vol. 7, Article ID 41729, 2017.

[18] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978-1979.

[19] M. Kitsak, L. K. Gallos, S. Havlin et al., "Identification of influential spreaders in complex networks," *Nature Physics*, vol. 6, no. 11, pp. 888–893, 2010.

[20] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[21] Y. Chen, G. Paul, S. Havlin, F. Liljeros, and H. E. Stanley, "Finding a better immunization strategy," *Physical Review Letters*, vol. 101, no. 5, Article ID 058701, 2008.

[22] A. Braunstein, L. Dall'Asta, G. Semerjian, and L. Zdeborová, "Network dismantling," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 113, no. 44, pp. 12368–12373, 2016.

[23] L. Zdeborová, P. Zhang, and H.-J. Zhou, "Fast and simple decycling and dismantling of networks," *Scientific Reports*, vol. 6, Article ID 37954, 2016.

[24] F. Morone and H. A. Makse, "Influence maximization in complex networks through optimal percolation," *Nature*, vol. 524, no. 7563, pp. 65–68, 2015.

[25] F. Morone, B. Min, L. Bo, R. Mari, and H. A. Makse, "Collective Influence Algorithm to find influencers via optimal percolation in massively large social media," *Scientific Reports*, vol. 6, Article ID 30062, 2016.

[26] L. Tian, A. Bashan, D.-N. Shi, and Y.-Y. Liu, "Articulation points in complex networks," *Nature Communications*, vol. 8, Article ID 14223, 2017.

[27] B. R. Da Cunha, J. C. González-Avella, and S. Gonçalves, "Fast fragmentation of networks using module-based attacks," *PLoS ONE*, vol. 10, no. 11, Article ID e0142824, 2015.

[28] S. Mugisha and H.-J. Zhou, "Identifying optimal targets of network attack by belief propagation," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 94, no. 1, Article ID 012305, 2016.

[29] A. Patron, R. Cohen, D. Li, and S. Havlin, "Optimal cost for strengthening or destroying a given network," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 95, no. 5, 2017.

[30] C. Xue-Qi, R. Fu-Xin, S. Hua-Wei, Z. Zi-Ke, and Z. Tao, "Bridgeness: a local index on Edge significance in maintaining global connectivity," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2010, no. 10, Article ID P10011, 2010.

[31] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. Election: Divided they blog," in *Proceedings of the 3rd International Workshop on Link Discovery (LinkKDD '05)*, pp. 36–43, ACM, New York, NY, USA, 2005.

[32] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[33] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proceedings of the KDD-2005: 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 177–187, ACM, New York, NY, USA, 2005.

[34] P. Erdös and A. Rényi, "On random graphs," *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.

[35] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Network robustness and fragility: percolation on random graphs," *Physical Review Letters*, vol. 85, no. 25, pp. 5468–5471, 2000.

[36] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, and T. Zhou, "Vital nodes identification in complex networks," *Physics Reports*, vol. 650, pp. 1–63, 2016.

[37] R. Lipton, D. Rose, and R. Tarjan, "Generalized nested dissection," *SIAM Journal on Numerical Analysis*, vol. 16, no. 2, pp. 346–358, 1979.

[38] I. A. Kovács and A.-L. Barabási, "Network science: destruction perfected," *Nature*, vol. 524, no. 7563, pp. 38-39, 2015.

[39] H.-J. Zhou, "Spin glass approach to the feedback vertex set problem," *The European Physical Journal B*, vol. 86, no. 11, article no. 455, 2013.

[40] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[41] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 65, no. 5, part 2, Article ID 056109, 2002.

[42] C. M. Schneider, A. A. Moreira, J. S. Andrade Jr., S. Havlin, and H. J. Herrmann, "Mitigation of malicious attacks on networks," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 108, no. 10, pp. 3838–3841, 2011.

[43] C.-K. Cheng and Y.-C. A. Wei, "An improved two-way partitioning algorithm with stable performance," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, no. 12, pp. 1502–1511, 1991.

[44] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

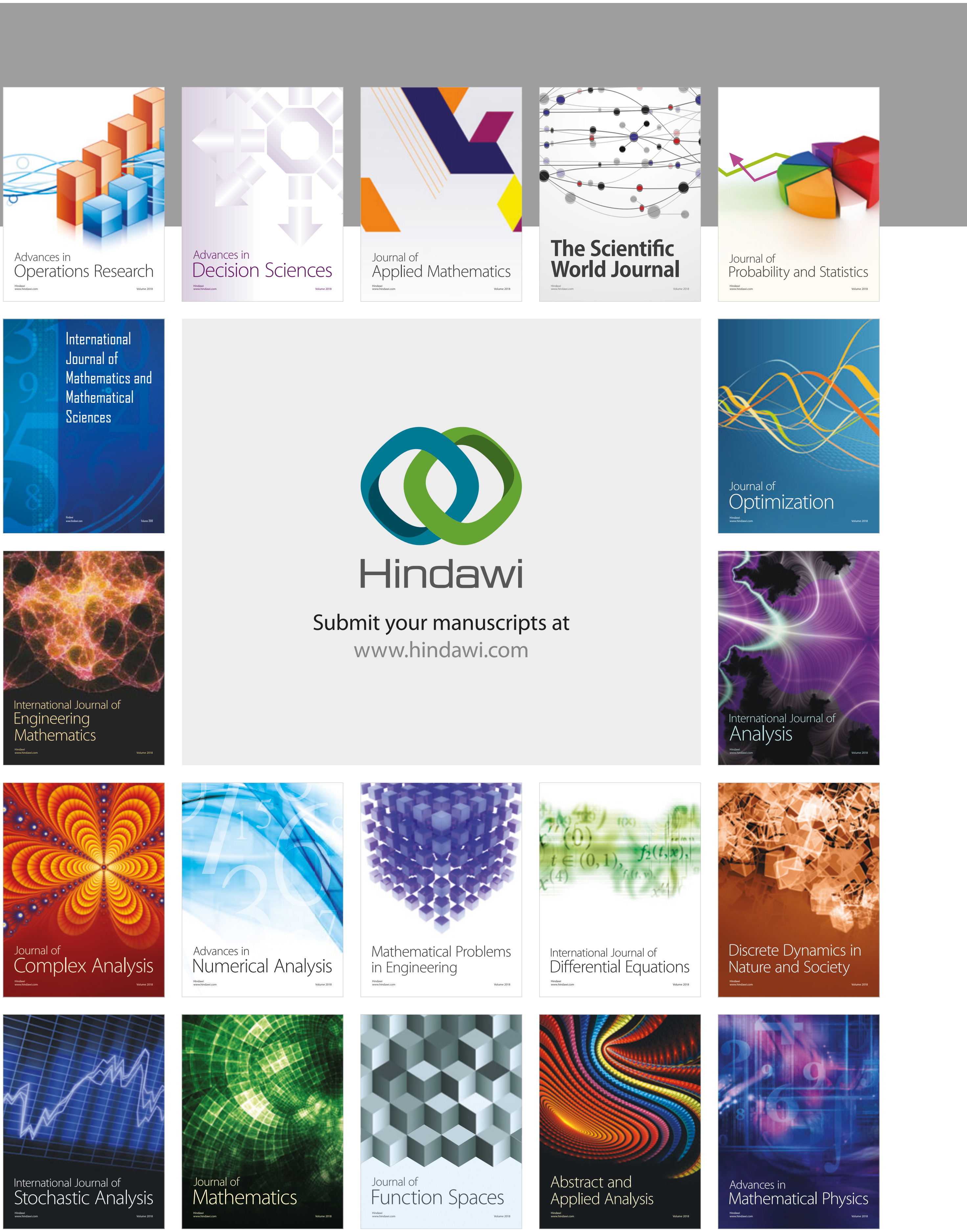
[45] H. Jia, S. Ding, X. Xu, and R. Nie, "The latest research progress on spectral clustering," *Neural Computing and Applications*, vol. 24, no. 7-8, pp. 1477–1486, 2013.

[46] J. Lurie, "Review of spectral graph theory," *ACM SIGACT News*, vol. 30, no. 2, p. 14, 1999.

[47] M. A. Riolo and M. E. J. Newman, "First-principles multiway spectral partitioning of graphs," *Journal of Complex Networks*, vol. 2, no. 2, pp. 121–140, 2014.

[48] R. Cohen and S. Havlin, *Complex Networks: Structure, Robustness and Function*, Cambridge University Press, Cambridge, UK, 2010.

[49] H. W. Hethcote, "The mathematics of infectious diseases," *SIAM Review*, vol. 42, no. 4, pp. 599–653, 2000.



Advances in
Operations Research
Advances in
Decision Sciences
Journal of
Applied Mathematics
The Scientific
World Journal
Journal of
Probability and Statistics
International
Journal of
Mathematics and
Mathematical
Sciences
Hindawi
Submit your manuscripts at
www.hindawi.com
Journal of
Optimization
International Journal of
Engineering
Mathematics
International Journal of
Analysis
Journal of
Complex Analysis
Advances in
Numerical Analysis
Mathematical Problems
in Engineering
International Journal of
Differential Equations
Discrete Dynamics in
Nature and Society
International Journal of
Stochastic Analysis
Journal of
Mathematics
Journal of
Function Spaces
Abstract and
Applied Analysis
Advances in
Mathematical Physics